REMARKS

Claims 1-4, 6-13, 15-21 and 23-37 are pending. Claims 5, 14 and 22 are cancelled.

Applicants acknowledge and appreciate that the Examiner has withdrawn the previous rejection of claims 1-4, 6-8, 16-21, 23-29, and 31-37 under 35 U.S.C. §102 in light of the arguments presented in the previous response to office action. New grounds of rejection in view of "Upgrading Microsoft Basic 6.0 to Microsoft Visual Basic.NET" ***(Robinson)*** and U.S. Patent No 6,230,313 ***(Callahan)*** are presented.

Claim Rejection – 35 U.S.C. §102

Claims 9-13 and 15 are rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 5,193,191 ***(McKeeman)***. Applicants respectfully traverse this rejection.

For ease of illustration and organization of arguments, remarks relevant to claims 9-13 and 15 are made in the claim 1 arguments section below.

Claim Rejection – 35 U.S.C. §103

The Examiner rejects claims 1-4, 6-8, 16-21, 23-29, 31-32 and 34-37 under 35 U.S.C. §103(a) as being unpatentable over ***McKeeman*** in view of ***Robinson***. Applicants respectfully traverse this rejection.

For ease of illustration, claim 1 is discussed first. Claim 1 calls for initiating compilation of a file in a processor-based system in advance of a request from a user to compile the file and detecting the user request to compile the file. Claim 1 also calls for indicating a status of the compilation of the file in response to detecting the user request. Initiating compilation of the file includes compiling the file in response to determining that the file has been modified.

The Examiner's rejection of claim 1 is improper because ***McKeeman*** and ***Robinson***, either alone or in combination as cited by the Examiner, fail to teach all of the claimed features. For example, claim 1 calls for initiating compilation of a file in a processor-based system in

advance of a request from a user to compile the file. In the current Office Action, the Examiner admits that ***McKeeman*** does not teach this feature, but the Examiner argues that ***Robinson*** teaches initiating compilation in advance of a request because ***Robinson*** teaches a background compiler that parses text after it is entered into a file. *See* Office Action, p.6 (citing ***Robinson***, p.16). The passage cited by the Examiner, however, does not teach or suggest initiating compilation of a file in a processor-based system in advance of a request from a user. ***Robinson*** teaches that a background compiler parses text as it is entered into a file in order to alert the user of syntax errors and the like. The Examiner's reliance on ***Robinson*** is misplaced, however, because the parsing is **not** the same as initiating compiling, as would be known to those skilled in the art. However, *arguendo,* even assuming that ***Robinson*** teaches a compiler that performs the parsing, an act of parsing text is just that: parsing text. ***Robinson*** does not disclose, and the Examiner has not cited, any teaching or suggestion in the cited reference that any parsing as shown in ***Robinson*** initiates compilation in **advance** of a request from a user to compile, as called for in claim 1. As such, ***Robinson*** does not, and cannot, teach this claim feature, and, as admitted by the Examiner, ***McKeeman*** fails to remedy the fundamental deficiencies of ***Robinson***.

Claim 1 also calls for indicating a status of the compilation of the file in response to detecting the user request. In the current Office Action, Examiner argues that ***McKeeman*** teaches this claimed feature. *See* Office Action, p.6 (now citing ***McKeeman***, col. 5, ll. 23-34). Applicants respectfully assert that the Examiner, as in the previous Office Actions, improperly characterizes the passages from column 5, lines 23-34 of ***McKeeman***. The Examiner states the cited passage of ***McKeeman*** teaches indicating a status of the compilation of the file in response to detecting the user request, because "errors are detected and reported." *See* Office Action, p.6 This position is untenable. ***McKeeman*** teaches that **after compiling** at least a portion of the file, errors in the file may be reported to the user. *See* ***McKeeman***, col. 5, ll. 23-34. In other words,

McKeeman teaches that errors in the file may be reported in response to attempting to compile the file and encountering errors. In contrast, claim 1 recites indicating a status of the compilation of the file in response to detecting the user request. The Examiner's attention is respectfully directed to the Specification for an illustrative, non-limiting example:

> "[I]n accordance with the present invention, because the task processing module 15 may have pre-processed one or more of the tasks associated with the build process, the task processing module 15, upon detecting (at 330) the user request to initiate the build, indicates (at 340) a status of the processing of the one or more tasks." *See* Specification, p.15, line 18 to p.16, line 4.

As this passage clearly illustrates, in the context of claim 1, compilation has already been initialized for at least a portion of a file, so the indication of the status of the compilation may be given in response to detecting the user request. ***McKeeman*** is not able to provide a status in response to detecting the user request, at least for the reason that there has not been any compiling done when the user input is received. That is, in ***McKeeman*** must detect a user request to compile, *then begin compiling* and then report errors in response to detecting errors during the compilation, which is in contrast to claim 1. It should be noted that any examples from the Specification are for illustrative purposes only, and do not limit the claims in any way.

When properly read in context, the cited passage in ***McKeeman*** does not teach that the status of the compilation is indicated in response to detecting a user input, as argued by the Examiner. The Examiner's arguments cannot be correct because the compilation in ***McKeeman*** has not yet taken place when the user request is received. In other words, ***McKeeman*** fails to teach or suggest indicating the status of the compilation in response to detecting a user input. As such, ***McKeeman*** does not, and cannot, teach the claimed feature of the status of the compilation is indicated in response to detecting a user input, as called for in claim 1. ***Robinson*** fails to remedy this fundamental deficiency as ***Robinson*** is concerned with providing users notifications of possible errors before any compiling is done; that is, ***Robinson*** is not concerned with any

relationships between compiling and providing status.

Claim 1 also calls for compiling the file in response to determining that the file has been modified. In the current Office Action, the Examiner cites ***McKeeman***, col. 5, ll. 21-23, as teaching this claimed feature. *See* Office Action, p.6. However, as Applicants have stated in the previous Response, ***McKeeman*** compiles when the user (developer) decides to compile, **not** in response to a file modification, as called for in claim 1. In the current Office Action, the Examiner argues that because unchanged files are not compiled, ***McKeeman*** teaches this claimed feature. The Examiner, however, has not shown how this passage teaches compiling the file in response to determining that the file has been modified, and this is not surprising because ***McKeeman*** teaches compilation occurs when the user (developer) decides to compile, **not** in response to a file modification, as called for in claim 1. As such ***McKeeman*** does not, and cannot, teach the claimed feature of the status of the compilation is indicated in response to detecting a user input, as called for in claim 1. ***Robinson*** fails to remedy this fundamental deficiency as ***Robinson*** is similarly concerned with compiling upon a user's command/input.

For at least the aforementioned reasons, claim 1 and its dependent claims are allowable. For at least similar reasons, the remaining independent claims, and their respective dependent claims are also allowable (including claim 9 and its dependent claims).

As such, Applicants request this rejection of claims 1-4, 6-13, 15-21, 23-29 and 31-37 under 35 U.S.C. §102(b) be withdrawn.

Claim 30 is rejected under 35 U.S.C. 103(a) as being unpatentable over ***McKeeman*** and ***Robinson***, and further above in view of U.S. Pat. Pub. No 2005/0108682 ***(Piehler)***. Applicants respectfully traverse this rejection.

Claim 30 depends indirectly from independent claim 24. Because ***McKeeman*** and ***Robinson*** fail to disclose all of the features of claim 24 (for at least the reasons discussed

earlier), these references likewise fail to teach the features of dependent claim 30. For at least this reason, claim 30 is allowable.

Claim 33 is rejected under 35 U.S.C. 103(a) as being unpatentable over ***McKeeman*** and ***Robinson*** and further in view of ***Callahan, II.*** Applicants respectfully traverse this rejection.

Claim 33 depends indirectly from independent claim 24. Because ***McKeeman*** and ***Robinson*** fail to disclose all of the features of claim 24 (for at least the reasons discussed earlier), these references likewise fail to teach the features of dependent claim 33. For at least this reason, claim 33 is allowable.

Arguments with respect to other dependent claims have been noted. However, in view of the aforementioned arguments, these arguments are moot and, therefore, not specifically addressed. To the extent that characterizations of the prior art references or Applicants' claimed subject matter are not specifically addressed, it is to be understood that Applicants do not acquiesce to such characterization.

In view of the foregoing, it is respectfully submitted that all pending claims are in condition for immediate allowance. The Examiner is invited to contact the undersigned attorney at (713) 934-4069 with any questions, comments or suggestions relating to the referenced patent application.

Respectfully submitted,

WILLIAMS, MORGAN & AMERSON, P.C.
CUSTOMER NO. 62293

Date: March 15, 2010

By: /Jaison C. John/
Jaison C. John, Reg. No. 50,737
10333 Richmond, Suite 1100
Houston, Texas 77042
(713) 934-4069
(713) 934-7011 (facsimile)
ATTORNEY FOR APPLICANT(S)